El Sistema CLS

Albert Pons Vega

Resumen– Cuando hablamos de las comunicaciones a través de internet, sabemos que hoy en día el uso de éstas pueden llegar a ser peligrosas a través de un uso indebido o inapropiado. Para ello se ha decidido llevar a cabo un estudio de los distintos protocolos más utilizados con el objetivo de diseñar un sistema de comunicación asíncrono con fines educativos. Dicho sistema se ha definido en base a tres protocolos; dos para la generación de claves entre usuarios y entre el servidor, y otro para el intercambio de mensajes asíncronamente. Posteriormente se ha realizado un análisis, repasando los principios de la seguridad de los datos, y evaluándolos individualmente con los puntos del sistema diseñado. Finalmente se han extraído las conclusiones referentes al sistema y se ha documentando todos los aspectos.

Palabras clave– Sistema, Protocolo, Handshake, Encriptación Simétrica, Encriptación Asimétrica, Cifrado

Abstract– When we talk about communications over the internet, we know that today the use of these can become dangerous through improper or inappropriate use. For this, it has been decided to carry out a study of the different most used protocols in order to design an asynchronous communication system for educational purposes. These system has been defined based on three protocols; two for the generation of keys between users and between the server, and another for the exchange of messages asynchronously. Subsequently, an analysis has been carried out, reviewing the principles of data security, and evaluating them individually with the points of the designed system. Finally, the conclusions regarding the system have been drawn and all aspects have been documented.

Keywords– System, Protocol, Handshake, Symmetric Encryption, Encryption Asymmetric, Cipher

1 Introducción

Hoy en día cada nuevo aspecto informático relacionado con la información que se incorpore al mercado, puede abrir las puertas a nuevas posibilidades de ataque. Algunos de estos casos se han podido demostrar a través de vulnerabilidades cómo aperturas en la confidencialidad de los datos, que se sitúan en los protocolos de comunicación. Es por eso que los autores como JJ Cano del estudio "Retos de seguridad/ciberseguridad en el 2030" han llegado a la conclusión de que la implementación de los protocolos de comunicación representan una gran parte de la seguridad que estos aplican, dado que son la clave para abrir una comunicación segura entre dos puntos. [1]

• E-mail de contacto: albert.ponsvega@gmail.com
• Mención realizada: Tecnologías de la Información
• Trabajo tutorizado por: Victor García Font (PREGUNTAR)
• Curso 2020/21

En este proyecto se ha diseñado un sistema de seguridad informático con fines educativos, para las comunicaciones asíncronas y de extremo a extremo [2], entre dos usuarios a través de una red. Para ello, se ha investigado los protocolos actuales más utilizados, como los protocolos TSL [3] y el protocolo TLS "handshake" [5] y los distintos métodos de encriptación simétrica y asimétrica. Posteriormente se ha diseñado y documentado todo lo necesario para entender el funcionamiento del sistema.

2 Objetivos

En este proyecto se busca desarrollar un sistema de seguridad orientado a las comunicaciones entre dos usuarios y analizar qué tan seguro o no es. Se centra en una comunicación asíncrona y de extremo a extremo, tal y como se ha mencionado anteriormente. Para ello se han determinado cuatro objetivos:

Protocolos para la generación de claves : Estudiar y diseñar todos los aspectos relacionados con el sistema en

cuestión y el intercambio de claves; como la generación de claves previa a la conexión entre un cliente y el servidor y la generación de claves entre dos clientes de la misma red.

Protocolo para el intercambio de mensajes : Diseño del protocolo para el intercambio de mensajes entre dos clientes de la misma red, de forma asíncrona y con una comunicación extremo a extremo.

Anàlisis teórico del sistema : Analizar el sistema teóricamente y sacar las conclusiones más oportunas referentes a los aspectos de seguridad: Confidencialidad, Integridad, Autenticidad, Disponibilidad y No Repudio [6].

Implementación del sistema* : Implementar a código el funcionamiento del sistema, juntamente con cada uno de los protocolos definidos.

* Éste objetivo es de baja prioridad.

3 Metodología

Para realizar este proyecto y llegar a conseguir los objetivos propuestos, se ha decidido usar Scrum, una metodología Agile [7]. Scrum es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto. Además se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto. Por ello, Scrum está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto.

En nuestro caso, esta metodología, se adapta bien aunque no se trate de un equipo de trabajo, ya que se busca diseñar un sistema con el fin de poder implementarlo y analizarlo; es por eso que se adecua el hecho de poder definir unos "puntos de entrega" para llevar el seguimiento de forma correcta y poder ver los avances.

4 Planificación del proyecto

Para la realización de este proyecto y la ejecución de cada 'sprint', hemos dividido el proyecto en distintas etapas:

ETAPA 0: Fase inicial de documentación sobre los distintos aspectos relacionados con el diseño del protocolo; como el funcionamiento de los protocolos handshake, criptografía simétrica y asimétrica, etc.

ETAPA 1: Definición y documentación del objetivo del sistema, juntamente con las distintas características que éste ofrecerá.

ETAPA 2: Definición del protocolo para la generación de claves entre un nuevo cliente y el servidor central de la red.

ETAPA 3: Definición del protocolo para la generación de claves entre dos clientes pertenecientes a la misma red, que deseen establecer conexión para el intercambio de mensajes.

ETAPA 4: Definición del protocolo para el intercambio de mensajes entre dos clientes de la misma red, asíncronamente y de extremo a extremo.

ETAPA 5: Análisis de la seguridad del sistema de forma global, analizando y detallando cada uno de los aspectos de la seguridad mencionados anteriormente.

ETAPA 6: Conclusiones y resultados finales del proyecto y la realización de éste.

En el Appendix A.1 figura 7 podemos encontrar el diagrama de gantt con los distintos periodos de tiempo estimados, para la realización del proyecto y con el fin de llegar a los objetivos propuestos.

5 Background

5.1 Criptografía simétrica y asimétrica

La criptografía es una herramienta muy útil para el cifrado de datos o información; cuando hablamos de criptografía informática, puede ser también entendida como un medio para garantizar las propiedades de confidencialidad, integridad y disponibilidad de los recursos de un sistema.

Principalmente encontramos la criptografía simétrica o de llave secreta [8]. Ésta es aquella criptografía que , a través de algún método matemático llamado sistema de cifrado, permite cifrar y descifrar un mensaje utilizando únicamente una llave secreta. De esta forma garantizamos que un mensaje 'm' cifrado con una llave secreta 'k' tan solo podrá ser descifrado con la misma llave. Esta llave debe compartirse entre los usuarios que se desee, para que ellos también puedan cifrar y descifrar. Un ejemplo de método criptográfico simétrico es AES o 3DES [9].

Fig. 1: Criptorafía Simétrica

Por otro lado encontramos la criptografía asimétrica [10]. Ésta permite cifrar y descifrar con distinta llave, es decir, en este caso no se usa una llave simétrica para cifrar y descifrar, sino que cada usuario dispone de una clave secreta y otra pública; con la que, para cifrar un mensaje, se cifra con la clave pública 'kX' del destinatario para que él pueda descifrar el mensaje con su clave secreta. Un ejemplo de método criptográfico asimétrico es RSA [11].

Fig. 2: Criptorafía Asimétrica

A la hora de comparar los dos tipos de cifrado, encontramos ciertas diferencias entre los dos métodos. La ventaja de los algoritmos de criptografía simétrica es que son rápidos, además, en los últimos años se han ido incorporando a los procesadores de ordenadores,

servidores, routers y otra serie de dispositivos la aceleración de cifrado por hardware. Aun así, el mayor enemigo de los sistemas criptográficos simétricos, y asimétricos, es el ataque por fuerza bruta, por lo que hace muy importante mantener el secreto de las claves compartidas y privadas.

Por otro lado, los algoritmos asimétricos, conllevan una desventaja frente a los algoritmos asimétricos; éstos no permiten un intercambio de claves para el cifrado de los mensajes dentro de un canal inseguro. Ésto se soluciona a través de la criptografía simétrica para el intercambio de claves. Aun así, tratan algoritmos de cifrado muy potentes; esto hace que el criptoanálisis de estos sistemas sea complicado y que los ataques de fuerza bruta realizados con computadoras actuales para romperlo resulte muy difícil. Es por eso que en este sistema se busca usar los dos protocolos, cubriendo uno las desventajas del otro [12].

5.2 Intercambio de claves Diffie Hellman

El protocolo criptográfico Diffie-Hellman, debido a Whitfield Diffie y Martin Hellman (autores también del problema de Diffie-Hellman o DHP), es un protocolo de establecimiento de claves entre partes que no han tenido contacto previo, utilizando un canal inseguro y de manera anónima (no autenticada) [13]. Un canal inseguro hace referencia a un canal de comunicación en el que un tercero puede interceptar los mensajes.

El sistema se basa en la idea de que dos interlocutores pueden generar conjuntamente una clave compartida sin que un intruso, que esté escuchando las comunicaciones, pueda llegar a obtenerla o interferir en ella. Para ello se eligen dos números públicos, es decir dos números conocidos, entre los usuarios que desean compartir la clave: el primer valor, corresponde a un número primo 'p' y el segundo a una base 'g'. Seguidamente cada uno de los usuarios genera un valor numérico secreto menor que 'p'. Éstos pueden ser un valor 'a' para un usuario A y un valor 'b' para un usuario B. A partir de aquí, cada uno de los usuarios aplica la siguiente función substituyendo el valor 'x' por los valores secretos 'a' y 'b':

$$A = g^a mod(p), B = g^b mod(p)$$

. A continuación los interlocutores se intercambian los resultados de forma pública. En teoría revertir esta función es tan difícil como calcular un logaritmo discreto (un sextillón de veces más costoso que la exponenciación usada para transformar los números). Por eso se dice que este número es el resultado de aplicar una función unidireccional al número secreto.

Finalmente ambos interlocutores utilizan por separado la misma fórmula utilizada anteriormente, pero substituyendo la base 'g' por el valor recibido del interlocutor opuesto, y el valor 'x' por el valor secreto de cada interlocutor. En el caso del usuario A sería:

$$K = (B)^a mod(p)$$

Por otro lado, para el usuario B:

$$K = (A)^b mod(p)$$

Con ésto, conseguimos que los dos lleguen al mismo número, que será la clave compartida.

Para más información consultar el protocolo Diffie-Hellman en la referencia [13].

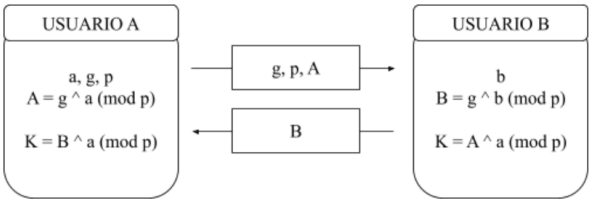

Fig. 3: Algoritmo Diffie-Hellman

5.3 Firma digital

Una firma digital, usado en la criptografía asimétrica, es una técnica matemática utilizada para validar la autenticidad e integridad de un mensaje, software o documento digital [15]. Normalmente se usa la clave privada para firmar la información y así, a través de la clave pública, poder verificarla. Además, se tiende a usar un resumen del documento a través de una función denominada "Hash". Esta función nos permite extraer un resumen unidireccional, es decir, un resumen imposible de revertir i extraer la información original. Además esta función nos permite tener una entrada de infinitos datos, mientras que el resumen de salida tendrá un tamaño fijo siempre.

A partir de aquí, la firma de un documento 'd' se realiza tomando un documento digital, se extrae el resumen del documento 'H(d)' y este resumen se cifra asimétricamente con la llave privada del firmante 'Ck1(H(d))', que representaría la firma digital, ahora hay que ponérsela al documento, para eso se concatenan el documento y su resumen cifrado. Entre otros, este sería un método para poder realizar la firma digital [16].

Finalmente y una vez transferida, se procede a verificar la firma digital, separando el documento 'd' del resumen y descifrando asimétricamente con la llave pública del emisor. De esta forma se obtiene el resumen del documento original H(d), el cual ya se está verificando su origen a través del uso de la clave pública para el descifrado. Seguidamente procede a realizar el mismo procedimiento que el usuario emisor, por lo que aplica la función hash al documento recibido 'd' y procede a comparar H(d) = H(d)'. En el caso que los dos resúmenes, el calculado y el recibido, sean el mismo, significa que la integridad sigue presente y los datos no han sido modificados.

6 ASUNCIONES

Para este proyecto se han determinado ciertas asunciones que hay que tener en cuenta. Estas permiten al sistema funcionar correctamente dado que hay valores utilizados durante los protocolos 6.1 y 6.2 que se requiere conocer de forma previa. Por ejemplo la clave pública del servidor o

el hecho de que se considere el servidor central como una entidad de confianza.

6.1 Servidor como entidad de confianza

Para que este sistema funcione correctamente requiere de, principalmente, un servidor o nodo central de confianza dado que él es el conocedor de las claves simétricas y los identificadores de usuario. Por lo tanto, al intentar establecer conexión con otro usuario, se debe confiar en que dicho servidor está asociando correctamente el usuario destino con la relación clave simétrica - Identificador, que él mismo tiene guardada. Estos identificadores serán generados por el servidor en el momento en el que un usuario finaliza de forma exitosa una generación de claves con el servidor; posteriormente, ese identificador es almacenado por el usuario e incluido en todos los paquetes.Es por eso que la primera asunción que se ha tenido durante el desarrollo del sistema es la presencia de un servidor fiable para los usuarios; es decir, un servidor en el que los distintos nodos lo perciben como una entidad de confianza.

6.2 Distribución de claves

En segundo lugar encontramos la distribución de claves. Esta asunción hace referencia al hecho de que cualquier usuario que desee conectarse a la red de un servidor en cuestión, debe disponer del valor de la clave pública de dicho servidor antes de iniciar ningún protocolo del sistema. Esto nos permite enviar datos al servidor desde el primer instante sin tener que realizar protocolos orientados al intercambio de claves como Diffie-Hellman o ElGamal. En el caso que no fuera así, se debería realizar uno de estos protocolos para poder intercambiarse las claves entre usuario y servidor, en un canal inseguro, des del primer instante.

Un canal inseguro hace referencia al hecho de que enviar información por internet sin cifrado ni métodos de ocultación de datos, permite a cualquier usuario atacar, tanto la privacidad, como la integridad y la autenticidad de los datos.

En este sistema también se usa Diffie-Hellman para extraer una clave simétrica entre un usuario A y otro usuario B; pero la diferencia es que en este sistema se realiza directamente en un canal seguro, porque se asume que la clave pública del servidor es un valor conocido por lo que permite evitar realizar dos veces el protocolo para el intercambio de llaves.

7 EL SISTEMA CLS

El sistema CLS, de las siglas “Chain Link System”, es un diseño propio de un Sistema de seguridad orientado a las comunicaciones entre dos dispositivos con el fin de garantizar una comunicación cifrada de extremo a extremo y de forma asíncrona.

Cuando hablamos de una comunicación cifrada de extremo a extremo, hablamos de aquel cifrado en el que tan solo los usuarios, o nodos, origen y destino, son capaces de descifrar el paquete de datos; aplicando así confidencialidad entre los dos nodos en cuestión, es decir, garantizando que los datos transmitidos se mantienen secretos para cualquier usuario externo a la comunicación, incluyendo el servidor.

Por otro lado, el hecho de que permita una comunicación asíncrona, hace referencia al hecho de que, aunque un usuario no esté conectado a la red, este pueda recibir el paquete de datos a posteriori, es decir cuando se conecte nuevamente. El servidor será el encargado de almacenar los paquetes en una cola de mensajes en el caso que el receptor no esté conectado. En este caso, cuando hablamos de paquetes hacemos referencia al conjunto de datos que se transmiten; primero desde un punto origen, pasando por un servidor y llegando a un punto destino.

A continuación se muestra la tabla de los valores representativos que se van a usar para la representación de los protocolos.

NOMBRE	DEFINICIÓN
PK-X	*Public Key* o Clave Pública del usuario X
SK-X	*Secret Key* o Clave Secreta del usuario X
SmK-XY	Clave simétrica (Entre usuario X y Y)
SEn(m, Y)	*Symmetric Enciphered* o Cifrado Simétrico de m con la clave Y
AEn(m, Y)	*Asymmetric Enciphered* o Cifrado Asimétrico de m con la clave Y
DS(m, SK)	*Digital Signature* o Firma Digital (de m) con la clave SK
verify(DS, PK)	Verificar la firma digital DS con la clave PK
m	Mensaje
P	Paquete
C	Paquete cifrado

Fig. 4: Tabla de valores representativos

En el caso de los descifrados, para la criptografía asimétrica, se usarán la misma representación que para el cifrado pero substituyendo la clave pública por la privada. Por otro lado, para la criptografía simétrica, al tratarse de la misma clave, se usará la misma representación.

7.1 Protocolo para la Generación de claves Usuario - Servidor

El primer protocolo del sistema, corresponde a la generación de claves, tanto asimétricas, como simétricas, entre un usuario y el servidor. Estas claves permitirán transmitir datos e información, entre dichos puntos, con un cifrado de extremo a extremo. Para este protocolo vamos a definir la clave pública del servidor como un valor conocido por los usuarios, al igual que se especifica en la sección de asunciones. Ésto nos permite evitar ataques de “Man in The Middle” que se explica con más detalle en la sección Análisis del sistema. Para la generación de claves entre el usuario y el servidor, se requiere de unos pasos descritos a continuación:

PASO 1 y 2: El usuario genera, primeramente sus claves pública y privada. Seguidamente genera un paquete ‘P’ con

una petición de conexión ‘Conn’, junto con su clave PK-A. Éste lo cifra con la clave pública del servidor PK-S y genera el paquete cifrado ‘C’. ‘Conn’ hace referencia a al parámetro que indica al servidor que se desea establecer una conexión.

$$P = [\text{‘Conn’}, \text{PK-A}]$$
$$C = AEn(P, \text{PK-S})$$

PASO 3: El servidor recibe el paquete, lo desencripta

$$C' = AEn(P, \text{SK-S})$$

y procesa la información. En este caso, al tratarse de una petición para una nueva conexión el servidor procede a la generación de una clave simétrica, compartida entre el usuario y el servidor SmK-AS. Seguidamente genera un paquete con dicha clave y la firma de la misma por el servidor. Finalmente lo cifra con PK-A. La razón por la que el servidor escoje la clave simétrica es por que se busca dar el poder de generar la clave al servidor, requeriendo de la confianza por parte del usuario, el solicitante de la petición.

$$P2 = [\text{SmK-AS}]$$
$$C2 = AEn(P2, \text{PK-S})$$
$$C3 = DS(C2, \text{SK-A})$$

PASO 4: El usuario recibe el paquete y procede a la realización del último paso: confirmar al servidor la recepción de la clave SmK-AS. Para ello genera un paquete con un mensaje de recepción. Para este caso y como ejemplo usaremos una variable llamada ‘AllRecived’. Este paquete lo cifra con la clave SmK-AS que el servidor le ha devuelto.

$$P3 = [\text{‘AllRecived’}]$$
$$C3 = SEn(P3, \text{SmK-AS})$$

En el caso que el servidory no reciba la información correctamente, éste esperará un periodo de tiempo y reenviará nuevamente el mensaje anterior con el objetivo de informar al usuario que no ha recibido la confirmación. Ésto se realizará hasta tres veces; a partir de aquí se enviará un mensaje al usuario diciendo que no ha sido posible establecer la confirmación, por lo que se requerirá reiniciar el proceso del protocolo des del principio. Por otro lado, el servidor debe disponer de algún sistema de errores para localizar posibles cambios por terceros usuarios malintencionados.

Independientemente de todo el seguimiento de pasos, para la siguiente sección 7.2, se debe remarcar que los paquetes que el usuario envíe hacia otro usuario perteneciente a la red, van a pasar por el servidor, por lo que dichos paquetes contiene un fragmento, cifrado con la clave simétrica extraída en esta sección, con la información referente al destino de del paquete.

7.2 Protocolo para la Generación de claves Usuario A - Servidor - Usuario B

En este escenario partimos de que los usuarios A y B ya han establecido conexión con el servidor y se mantienen a la espera de que alguno de los dos decida comunicarse con el otro. Esto implica que cada uno de los usuarios dispone del conocimiento de 3 claves:

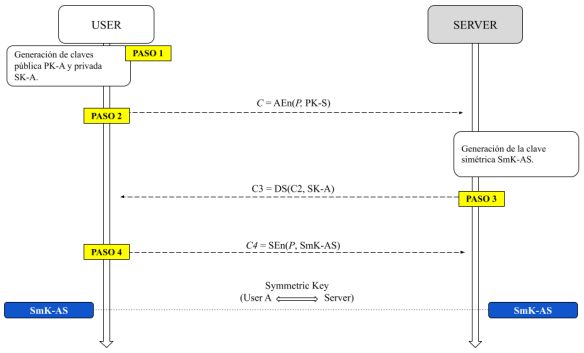

Fig. 5: Generación de claves Usuario-Servidor

SmK-AS/BS : Clave simétrica compartida entre el usuario A/B y el Servidor.
PK-A/B : Clave pública del mismo usuario A/B.
SK-A/B : Clave privada del mismo usuario A/B.

En este caso supondremos el escenario en el que el usuario A se dispone a establecer conexión con el usuario B. Para ello se busca seguir los mismos pasos aplicados en la conexión Usuario - Servidor, con el fin de generar la clave simétrica SmK-AB; esta vez, pero, aplicado a un tercer usuario B.

PASO 1: El usuario A, desea poder comunicarse con el usuario B de tal forma que los datos enviados tan solo sean leídos por él. Además queremos garantizar que nadie más influye en la generación de la clave, más que ellos dos (Usuario A y B);

Para ello los dos usuarios deberán encontrarse, o deben ser encontrados, físicamente para transmitirse sus claves públicas a través de algún mecanísmo de corta fidelidad como NFC (Near Field Communication), que se trata de una tecnología inalámbrica que funciona a través de frecuencia y con un rango no superior a los 20 centímetros. Además de sus claves públicas, también se compartirán los identificadores individuales que el servidor ha generado anteriormente para cada uno de ellos.

Este proceso nos permite evitar intrusiones de terceros en la generación de claves entre usuarios, garantizando así, cifrado desde el primer instante.

PASO 2: Una vez estos dos usuarios hayan intercambiado sus claves públicas, el usuario A encripta una solicitud de clave simétrica con el usuario B, juntamente con un fragmento de llave SmK-AB[½] y su clave PK-A. Este fragmento SmK-AB[½] ha estado generado a través del protocolo Diffie-Hellman, orientado a la generación de claves simétricas entre dos o más usuarios; con lo que corresponde a los valores necesarios para la generación de dicha clave simétrica SmK-AB. Finalmente, cifra el paquete y lo firma.

$$P = [\text{‘SmK?’}, \text{SmK-AB}[½]]$$

C = AEn(P, PK-B)
C2 = DS(C, SK-A)

El servidor, al tratarse de un intermediario en estas comunicaciones, recibe el paquete, procesa la información requerida y procede a reenviarlo al usuario B. En el caso que el usuario B no esté conectado, el servidor, al no recibir respuesta del usuario B, procederá a guardar en la cola de mensajes dicho paquete para que, posteriormente, el usuario B pueda realizar una bajada de mensajes al establecer nuevamente conexión. Por otro lado el servidor, mantendrá un registro de identificadores relacionados con las claves simétricas de los usuarios con los que haya establecido conexión, para que posteriormente tener el conocimiento para el reenvío de paquetes.

PASO 3: Seguidamente, y con el conocimiento de la dirección a la que va ese paquete, procede a enviarlo al usuario en cuestión; en este caso el usuario B. Este paquete no recibe ninguna modificación.

PASO 4: El usuario B recibe la petición y pasa a procesarla. Primeramente comprueba que la firma esté correcta,

verify(DS, PK-A)

y que los datos no hayan sido alterados. Seguidamente continúa con el proceso y, para ello, calcula su fragmento de clave SmK-AB[2/2], a través del mismo protocolo Diffie-Hellman, y procede a generar un nuevo paquete con dicho fragmento y su clave PK-B. A éste, se le añade la firma del mismo y se cifra todo con la clave pública PK-A.

P2 = [SmK-AB[½], PK-B]
C3 = AEn(P2, PK-A)
C4 = DS(C3, SK-B)

PASO 5: Seguidamente el servidor recibe el paquete y procede a reenviarlo hacia el usuario A de la misma forma que lo ha realizado anteriormente para el usuario B, pero con los cambios pertinentes.
A partir de este punto se da por hecho que el usuario B ya dispone de la clave simétrica entre A y B SmK-AB dado que ya conoce las dos partes requeridas para la generación de clave a través del protocolo Diffie-Hellman.
PASO 6: El usuario A recibe los datos y comprueba la firma, al igual que ha realizado el usuario B anteriormente. A continuación, calcula, de la misma manera y con los mismos fragmentos, la clave simétrica SmK-AB, y procede a generar un paquete con un mensaje de conexión establecida con éxito. En este caso usaremos de ejemplo el mensaje 'Connection Successful'. Éste paquete lo cifra con la clave simétrica calculada y se lo manda al servidor para que este se lo entregue al usuario B.

P = ['Connection Successful']
C = SEn(P, SmK-AB)

PASO 7: Seguidamente el servidor recibe el paquete y procede a reenviarlo hacia el usuario B de la misma forma que lo ha realizado anteriormente para el usuario A, pero con los cambios pertinentes.

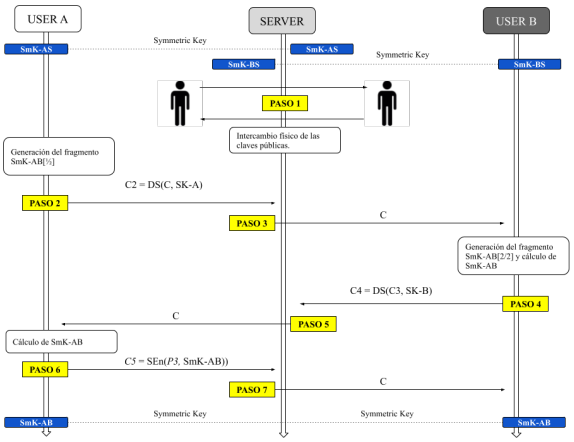

Fig. 6: Generación de claves Usuario A - Usuario B

7.3 Protocolo para el intercambio de mensajes

En este protocolo partimos del hecho de que los usuarios A y B ya han establecido conexión y disponen de una clave simétrica compartida SmK-AB. A partir de aquí se seguirá un seguido de pasos para el intercambio de mensajes entre usuarios.

Para ello se usará un método de colas para mensajes, para que los distintos usuarios realicen las comunicaciones de forma asíncrona. Una cola de mensajes es una forma de comunicación asíncrona de servicio a servicio que se usa en arquitecturas de microservicios y sin servidor. Los mensajes se almacenan en la cola hasta que se procesan y eliminan. De esta forma los usuarios podrán enviar los mensajes de forma habitual, y los destinatarios, en caso de estar desconectados, podrán descargárselos en su próxima conexión.

PASO 1: El usuario A se dispone a enviar un mensaje P al usuario B. Éste lo cifra con la clave simétrica SmK-AB, a lo que seguidamente añade la información P' necesaria para que el servidor pueda procesar la dirección destino del paquete. Esta se cifra con la clave simétrica SmK-AS y lo concatena con el mensaje formando un único paquete M.

P = SEn(P, SmK-AB)
P' = SEn(P, SmK-AS)
M = P + P'

PASO 2: El servidor recibe el paquete y al procesarlo procede a reenviarlo hacia el usuario destino. Para ello descifra la parte correspondiente del mensaje que va dirigida a él y procesa la direrección destino a la que tiene que ejecutar el envío. Seguidamente procede a comprobar que el usuario destino esté conectado.
P = SEn(P, SmK-AB)
En el caso que no lo esté, guardará el mensaje hasta la próxima conexión por parte del usuario B, que una vez conectado, realizará un pull para bajarse los mensajes pendientes de recepción. En el caso que sí esté conectado procederá a reenviar el mensaje a trarvés de la tabla de referencias entre claves simétricas e identificador; de esta manera el servidor sabe a quién debe reenviar el paquete.

Por otro, un usuario desconectado, al retomar la conexión se descargará todos los mensajes que estén almacenados en la cola para poder leerlos.

8 ANÁLISIS TEÓRICO DEL SISTEMA

En este apartado se realizará un análisis teórico del sistema diseñado. Para ello se analizarán los distintos principios de la seguridad de la información mencionados anteriormente y que podemos encontrar en detalle en las referencias del documento.A continuación se repasan los principios de la seguridad de la información mencionados anteriormente. Se analizará el sistema en base a los principios.

8.1 Confidencialidad

En este sistema, dato mencionado en la sección 6 Asunciones, partimos del hecho de que disponemos del conocimiento de la clave pública del servidor. Esto nos permite comunicarnos con él de forma cifrada desde el primer instante, es decir: cualquier usuario con intenciones de establecer conexión con el servidor, deberá realizar la primera petición y, como dispone de la clave pública de éste, puede adjuntar la suya en esta primera petición. A partir de ahí los dos ya disponen de las claves públicas del otro, por lo que se pueden comunicar de forma segura y garantizando la confidencialidad en los datos desde el primer momento.

Un ataque ya conocido que se podría aplicar a la confidencialidad sería, por ejemplo, el MitM o mejor conocido como “Man in the middle”. Este consistiría en posicionarse en medio de una comunicación, haciéndose pasar por servidor y usuario a la vez. De esta manera recibe los datos del servidor y del usuario, los lee, y los envía al destinatario con la posibilidad de que los haya modificado. Esto es debido a que la primera petición de un usuario hacia el servidor va en plano, por lo que este tercer usuario puede interferir en el envío y hacerse pasar por el servidor; y de la misma manera pero al revés. De todas formas requiere comentar que este ataque también estaría atentando contra la autenticidad de los usuarios y del servidor, ya que alguien se estaría haciendo pasar por otro que no és.

Aquí, en el sistema CLS, no es posible realizar este ataque dado que ya disponemos del conocimiento de la clave pública (Sección 6 Asunciones). En el caso que un usuario desease atacar la confidencialidad de los datos debería conocer los valores de las claves privadas con el fin de descifrar el mensaje, leerlo y modificarlo en el caso que se desease. O, por otro lado, debería conocer los valores de las claves simétricas que comparten los distintos usuarios para poder realizar el mismo proceso.

En el caso que se llegara a filtrar una clave, el usuario malicioso podría llegar a hacerse pasar por uno de los propietarios de dicha clave, o simplemente escuchar los mensajes transmitidos. Dado el caso, el usuario aplicaría un reinicio de las claves y las volvería a generar nuevamente, informando al usuario destino, con el cual se comparte la información de la clave, que la calve que compartían hasta ahora queda desautorizada y que se procede a realizar una nueva generación.

Aún todo lo citado anteriormente, el hecho de que los usuarios deban encontrarse físicamente para poderse intercambiar las claves públicas a través de algún sistema cómo NFC hace realmente dificil a un tercer usuario interferir en el proceso de generación de claves simétricas entre usuarios. Esto es debido a que, al ya disponer de la clave pública del usuario obtenida físicamente, todos los paquetes intercambiados des del primer instante irán cifrados, por lo que ataques como el mencionado anteriormente no se ve factibles de aplicar.

8.2 Autenticidad

En este sistema, la autenticidad de los datos recae en la información que se transmiten los distintos usuarios, garantizando así que los datos recibidos por un receptor corresponden a los datos enviados por el emisor original. Dado que confiamos en el servidor y se realizan intercambios de información constantemente cifrados, la generación de claves simétricas no se debería ver afectada, por lo que la autenticidad se cumple a través del cifrado. Es decir si tan solo dos usuarios disponen de una clave simétrica para cifrar y descifrar mensajes, cualquier mensaje recibido proveerá del usuario opuesto. Además, cada cierto período de tiempo se realizará nuevamente la generación de claves entre usuarios para comprobar, a través de la clave pública, que el usuario en questión es quien dice ser.

Por otro lado, aligual que en la confidencialidad, el hecho de que los usuarios deban encontrarse físicamente para intercambiarse las claves públicas, requiere de una confianza por parte de los usuarios en cuestión. A partir de aquí, y una vez intercambiadas las claves, en el caso que se filtrara una de las claves públicas, el usuario afectado por la filtración podría averiguar de quién se ha tratado el acto de distribución o filtrado.

8.3 No-Repudio

Para el principio de no repudio o irrenunciabilidad, encontramos el hecho de que todos los usuarios deben pasar por el servidor cuando desean enviar datos, por lo que el servidor, una entidad de confianza, debe conocer de dónde y hacia dónde va la información transmitida; por lo que ya podemos garantizar que un mensaje recibido, proviene de otro que, anteriormente se lo ha mandado al servidor, y éste, a través del identificador, sabe de quién viene y a dónde va. Es decir, dicho usuario emisor del mensaje no podrá negar el hecho de haber mandado el mensaje. Además y siguiendo el principio de la integridad de los datos, se entiende que las claves simétricas tan solo son compartidas por dos únicos usuarios, por lo que un mensaje descifrable con una clave en concreto debe provenir del único usuario opuesto.

Por otro lado, el hecho de estar usando firmas digitales en la generación de las claves simétricas y durante el intercambio de mensajes, hace que, un usuario receptor pueda comprobar que un mensaje proviene de alguien y

que éste ha estado firmado a través de su clave privada, por lo que no puede negar haberlo enviado.

Además, al igual que se explica en la sección de 8.2 Autenticidad también se puede implementar un sistema de identificación (a través de un usuario y una contraseña) para que los distintos usuarios se identifiquen en la red, por lo que así se facilita el registro de los paquetes transmitidos, dado que el servidor personalmente a cedido el acceso a cada usuario a través de su identificación y autenticación.

8.4 Integridad

Cuando se realiza una modificación sobre unos datos de forma no autorizada se considera que se está atacando a la integridad de los datos, es decir, que la información puede estar falsificada o manipulada por un tercer participante. En este caso, y en las comunicaciones en general, un claro ejemplo trataría un mensaje enviado por un emisor, interceptado y modificado por un usuario ajeno a la comunicación, y recibido finalmente por el receptor. Éste último, no estaría recibiendo la información original enviada, dado que ha sido modificada; es por eso que se considera un ataque contra la integridad de los datos.

En el sistema CLS la integridad de los datos no se puede ver afectada por usuarios ajenos dado que la información va cifrada constantemente. En el caso que dicho usuario desease interferir y modificar los datos con el fin de alterarlos, debería conocer la clave simétrica que dichos usuarios comparten. En caso contrario, cualquiera de los extremos (origne o destino) se dará cuenta de que dicho mensaje a sido modificado. Para ello debería ejecutar métodos conocidos para la extracción de claves, como por ejemplo, la fuerza bruta. Éste consiste en probar valores de clave hasta encontrar el valor correcto. Este valor representará la clave simétrica usada entre los usuarios que están manteniendo una conversación.

Además, del mismo modo que en el apartado anterior, la firma digital nos permite garantizar que los datos no han sido modificados.

En este sistema se utiliza criptografía asimétrica para establecer la clave simétrica; con lo que posteriormente, se usará la clave simétrica para la comunicación. En el caso de la clave asimétrica, existen protocolos criptográficos que usan valores de hasta 2048 bits, lo que supone un número de unas 300 cifras en decimal y miles de años de cómputo para el ordenador [17]. Este es el truco de los algoritmos de clave pública, es decir, crear un problema relativamente fácil de plantear pero muy costoso de resolver. Un ejemplar sería RSA. Contrariamente y para la clave simétrica, con un valor de 192 bits o 256 bits y algoritmos como AES, se tardarían millones de años en sacar el valor de la clave. [18]

Por otro lado y del mismo modo, el servidor tampoco puede conocer los datos que pasan a través de él, ya que no dispone del conocimiento de la clave que los usuarios en cuestión comparten.

8.5 Disponibilidad

En último lugar tenemos la disponibilidad en el sistema CLS. Esta no se cumple ya que todo el sistema depende de un servidor central. Dado que los mensajes, las conexiones y todos los datos prácticamente pasan por el servidor, y este tiene la función de re-enviarlos.

Dicha disponibilidad consiste en ofrecer el servicio siempre que el usuario lo desee, pero en este caso, cualquier usuario ajeno a la red con fines maliciosos, puede ejecutar ciertos procesos para provocar la caída del servidor. Un claro ejemplo sería realizar un DoS o mejor conocido como “Denial of Service”, que consiste en, a través de multitud de solicitudes, provocar el colapso de un servidor. Esto pasa ya que un servidor dispone de un límite de peticiones que puede procesar; por lo que cuando el número de peticiones supera a este límite, el servidor puede quedar colapsado y dejar de estar operativo.

En el caso que un usuario o un grupo de usuarios denegara el servicio del servidor, éste no podría responder ni procesar las peticiones recibidas, por lo que no se enviarían ni recibirían datos. Es decir, el principio de la disponibilidad no se cumple.

Alguna idea alternativa que se ha planteado como solución, sería distribuir el servidor en distintos nodos, formando una red Peer to Peer híbrida, donde algunos usuarios fueran también servidores. Una red Peer to Peer híbrida es una tipo de red donde los usuarios se encuentran conectados entre ellos y distribuidos entre usuarios y servidores/usuarios. O por otro lado, y de forma más directa, distribuir el propio servidor en distintos nodos para que, cuando uno se vea afectado por una denegación de servicio, se disponga de los otros para seguir ofreciéndolo.

9 Conclusions

Una vez acabado y analizado el proyecto hemos podido ver ciertos aspectos de la seguridad informática muy importante, juntamente con la necesidad de que todo sistema y/o protocolo tiene que estar muy bien diseñado y estudiado para garantizar que todos los principios de la seguridad informática se cumplan correctamente.

Otro aspecto que hemos podido apreciar es la necesidad de implementar una metodología para que los usuarios puedan comunicarse de forma directa, sin tener que pasar por un servidor central. Ya que como hemos podido ver, al requerir de un servidor central que haga de distribuidor de paquetes, puede afectar a la disponibilidad del sistema a través de cualquier ataque de denegación de servicio.

Por otro lado, se ha llegado a la conclusión de que diseñar un sistema o protocolo de seguridad para las comunicaciones es una tarea muy difícil, dado que se requiere de un conocimiento muy elevado, de muchos aspectos de la informática como: las matemáticas para el tratado de los valores, las redes de comunicación juntamente con los conceptos de internet, etc. Es por eso que este proyecto se ha diseñado un sistema con fines

educativos, con el objetivo principal de analizar y aprender más a fondo sobre la seguridad informática.

Referente al proyecto, se ha podido llevar a cabo los objetivos planeados al principio de éste. Se han podido definir todos los aspectos relacionados con el sistema en cuestión, es decir, los tres protocolos que lo forman; la implementación de un cifrado extremo a extremo con una comunicación asíncrona, y finalmente el análisis teórico de los aspectos de la seguridad de la información. Por otro lado, el objetivo de implementación, no se ha realizado y se ha dejado para realizar en el futuro.

En conclusión, los protocolos de seguridad son muy importantes; el hecho de que se transmita información a través de internet requiere de una base muy estudiada y bien pensada, para que los datos transferidos no se vean afectados, ni alterados. Es decir estén seguros ante amenazas de terceros. Es por eso que no se puede diseñar un protocolo de comunicación a la ligera sin analizarlo y estudiarlo adecuadamente; al intervenir tantos elementos de diferentes áreas en una comunicación, y todos los factores que pueden interferir en dicha comunicación, hace que el diseño de un sistema y/o protocolo resulte de una tarea muy precisa con mucha observación en cada paso que se realiza, ya que algun paso puede llegar a ser inecesario o inseguro por culpa de un detalle inpredecible que se ha pasado por alto a la hora de diseñarlo.

Agradecimientos

En primer lugar, me gustaría agradecer a Sergi Robles, coordinador de la mención, por haberme ayudado a descubrir lo que realmente me interesa, que son las distintas tecnologías de la información y los distintos aspectos que las envuelven. En segundo lugar, me gustaría agradecer a Víctor García, tutor del trabajo de final de grado realizado durante el curso 2020/2021, por haberme guiado y aconsejado en las decisiones que se han ido tomando durante el desarrollo del proyecto. Y, finalmente, quiero agradecer a todos los profesores que han pasado por mi enseñanza universitaria y que han hecho posible la formación de la que actualmente dispongo.

References

[1] JJ Cano (2020). Retos de seguridad/ciberseguridad en el 2030. Sistemas, 2020 - sistemas.acis.org.co.

[2] http://en.wikibooks.org/wiki/LaTeX

[3] K Hickman, T Elgamal. The SSL protocol: 1995 - webstart.com

[4] M Marlinspike, T Perrin. The x3dh key agreement protocol: Open Whisper Systems, 2016 - signal.org

[5] P Morrissey, NP Smart, B Warinschi. The TLS handshake protocol: A modular analysis. Journal of Cryptology, 2010 - Springer

[6] O Blancarte. Confidencialidad, integridad y autenticidad en mensajes. 2015 - orscarblancarte.org

[7] Apoorva Srivastava, Sukriti Bhardwaj, Shipra Saraswat. SCRUM model for agile methodology: Conference on Computing ..., 2017 - ieeexplore.ieee.org

[8] GJ Simmons. Symmetric and asymmetric encryption: ACM Computing Surveys (CSUR), 1979 - dl.acm.org

[9] YTM Vargas, HAM Mnedez - Mundo Fesc, 2015 - dialnet.unirioja.es Comparación de algoritmos basados en la criptografía simétrica DES, AES y 3DES

[10] GJ Simmons. Symmetric and asymmetric encryption: ACM Computing Surveys (CSUR), 1979 - dl.acm.org

[11] E Milanov. The RSA algorithm,: RSA Laboratories, 2009 - pdfdirectory.com,

[12] M Bellare, P Rogaway - Optimal asymmetric encryption : Workshop on the Theory and Application of of ..., 1994 - Springer

[13] UM Maurer, S Wolf . The diffie–hellman protocol: Designs, Codes and Cryptography, 2000 - Springer

[14] W Tzong-Chen, S Hung-Sung - Computers Security, 1996 - Authenticating passwords over an insecure channel Elsevier

[15] Introducción a la criptografía - Firma digital: 2006 - ru.tic.unam.mx

[16] S Wilson - Information Management Computer Security, 1999 - emerald.com

[17] P Mahajan, A Sachdeva. A study of encryption algorithms AES, DES and RSA for security: Global Journal of Computer ..., 2013 - computerresearch.org,

[18] A Biryukov, D Khovratovich. Related-key cryptanalysis of the full AES-192 and AES-256: International conference on the theory and ..., 2009 - Springer

Apéndice

A.1 Diagrama de Gantt

En este apéndice encontramos el diagrama de gantt referente a las etapas del proyecto explicadas en el punto 4, Planificación del proyecto.

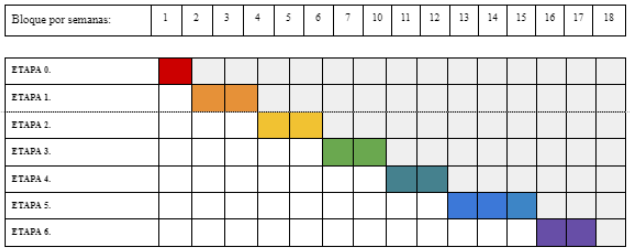

Fig. 7: Diagrama de Gantt de la planificación